Video Processing for Train Bottom View

Project Overview

The aim of this project is to process side-view videos of trains, detect and count each coach, and generate structured outputs including per-coach clips, representative frames, and reports. The project combines classical computer vision techniques and deep learning-based object detection to achieve accurate results.

Initial Approach:

Frame Extraction and Manual Counting:

- Started with a basic method of reading the video frame-by-frame using OpenCV.
- Extracted images and attempted to manually count coaches by analyzing frames.
- This method predicted **14 coaches** but the ground truth was **54 coaches**.
- It was tedious and highly error-prone due to occlusions, repetitive/near-identical coach appearance, and viewpoint issues.

Transition to Pre-Trained YOLO Model:

- Implemented a pre-trained YOLOv8 model (medium, yolov8m.pt) for automatic detection.
- Initial results were not accurate: the model missed multiple coaches and engines.
- The pretrained model initially predicted **0 coaches**. Attempts to force detections (e.g., lowering thresholds) produced many false positives — up to **~900** detections — because the pretrained model is trained to detect whole trains rather than individual coaches.
- Conclusion: the pretrained weights are unsuitable for coach-level detection without fine-tuning on coach annotations.

Dataset Preparation:

- Downloaded approximately 450 images from the train videos.
- Annotated the images using LabelImg with classes: coach and engine.
- Ensured that all label files were correctly formatted and consistent with the YOLO class configuration.

Model Training:

- Fine-tuned YOLOv8 medium (yolov8m.pt) on the annotated dataset.
- Training took approximately 30 minutes on Colab with a Tesla T4 GPU.
- Initial detection threshold was set at 0.35, which resulted in poor detection performance.

Threshold Tuning and Accuracy Improvement:

- Experimented with different confidence thresholds to improve detection results.
- Adjusted the threshold to 0.64, which produced significantly more accurate results, detecting 49 coaches in the sample video.
- Increased workers=2 to balance training efficiency and data loading speed.

Video Processing Pipeline

- Developed a complete detection and tracking pipeline using the fine-tuned YOLOv8 model.
- Annotated video with bounding boxes for coaches and engines.
- Organized outputs into structured folders for easy access and reporting.

Key Features

- Automated coach and engine detection with YOLOv8.
- Adjustable detection thresholds for fine-tuning performance.

Notes & Observations

- Using pre-trained YOLO alone is insufficient for accurate coach detection due to occlusions and perspective.
- Fine-tuning with a dedicated dataset of 450 annotated images significantly improved performance.
- Confidence thresholds play a critical role in balancing false positives vs. missed detections.
- GPU acceleration (Colab Tesla T4) is highly recommended for training and video processing.